A programmable audio DSP filter

Introduction

Trying to pull an individual signal out from adjacent frequency interference or other similar signals within the audio bandwidth of the receiver can often be frustrating. Almost all new radios have built-in techniques for separating such signals and letting you focus in on the desired transmission. Filtering techniques are applied to both RF and audio stages in order to achieve the best results.

Many older radios either have analogue filters or, in a few cases, no filtering at all. Whilst it would be challenging to retrofit appropriate filters into the RF stages of such radios, much can be done to improve the separation and readability of signals using audio-stage filtering techniques.

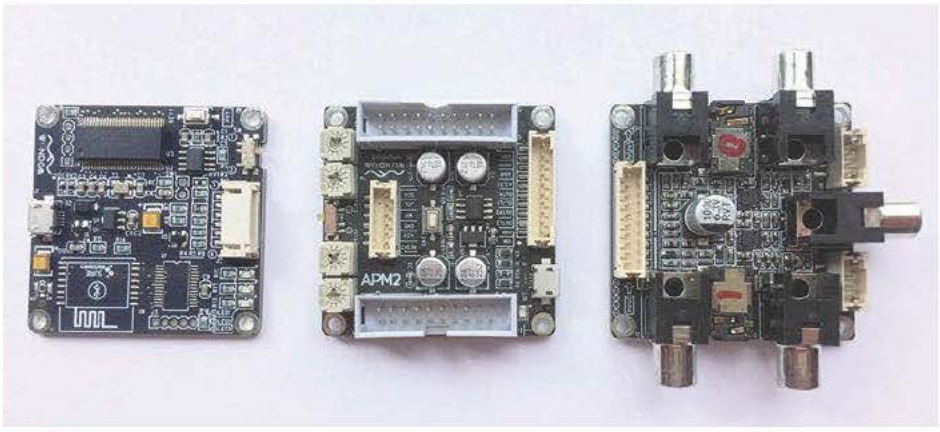

PHOTO 1: (L to R) USB programmer, processor board and audio interface board.

The analogue approach

Analogue audio filters are relatively easy to make for particular frequencies although they suffer from two main disadvantages. In order to change the frequency of an analogue filter, component values need to be changed; this can make a multi-frequency or multi-bandwidth filter complex and/or involve duplicating circuits. Secondly, analogue filter roll-off characteristics – selectivity, if you like – is generally pretty poor for single stage analogue filters. Properly designed multiple stages can improve this, but component count rises quite quickly with performance and the precision needed for each component value can make the costs mount up.

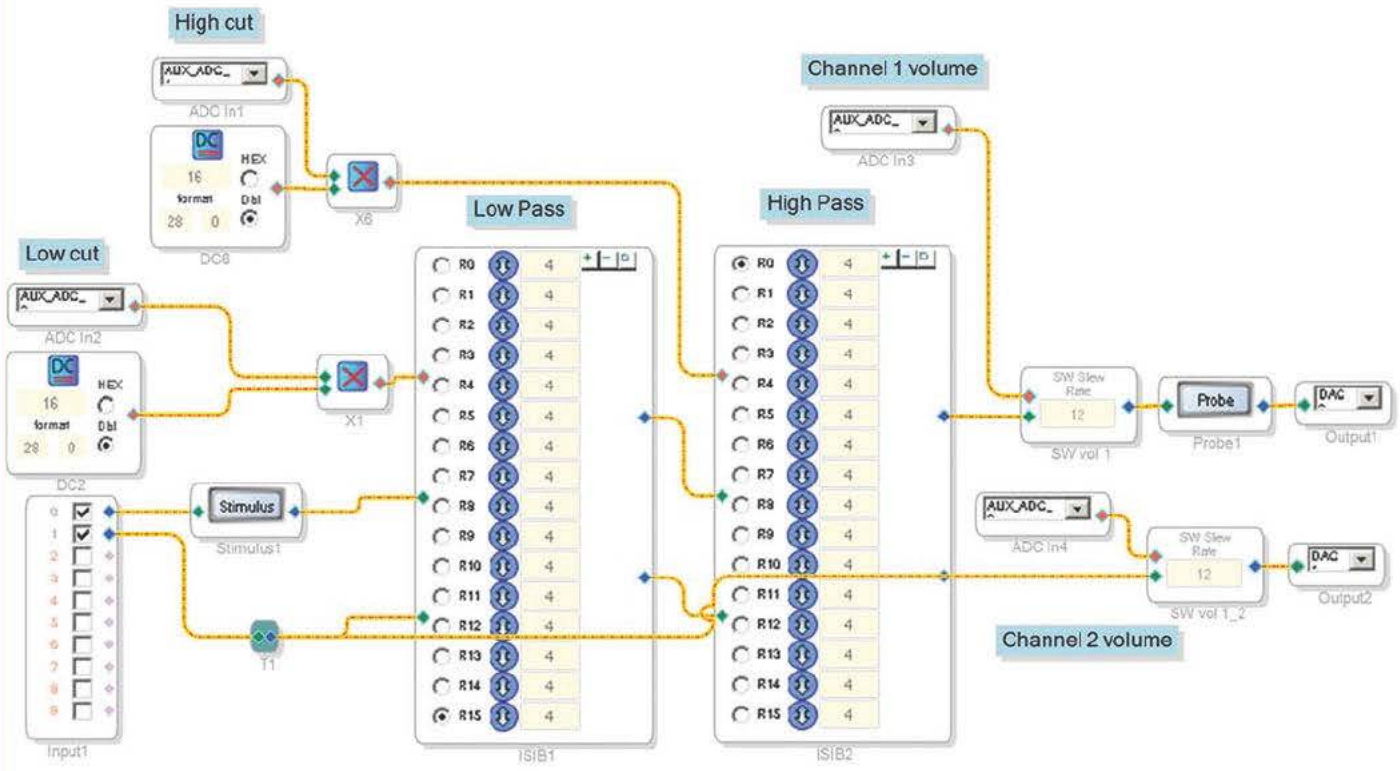

FIGURE 1: How the DSP filter design is represented in Sigmastudio.

FIGURE 2: Example filter settings.

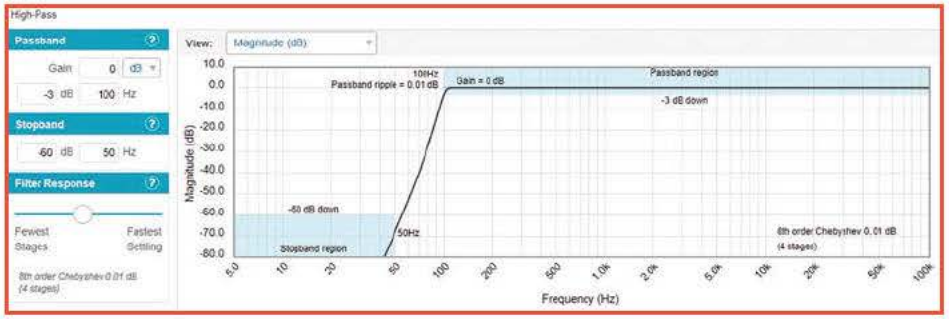

FIGURE 3: Magnitude plot from filter wizard.

Digital filters

Digital signal processing (DSP) filters use an analogue to digital converter (ADC) to change the signal into numbers, then the processing is achieved using mathematical functions (this month's Design Notes has a little more on the theory). Digital filters have the advantage that more-complex filters simply require a bit more maths, ie processing, on the signal. Once the numbers have been crunched, they're turned back into analogue signals using a digital to analogue converter (DAC).

This article describes a digital multiband audio filter that simulates sixteen 4-stage analogue high pass audio filters and sixteen 4-stage analogue low pass audio filters. This provides audio band pass capabilities from 100Hz across most of the usable communication audio spectrum and enables you to isolate tiny segments of the audio spectrum. Unlike some of the commercially available versions, this design does not limit you to a particular frequency of audio operation.

The filter is suitable for CW, SSB and digital modes and uses a low cost DSP module and free software.

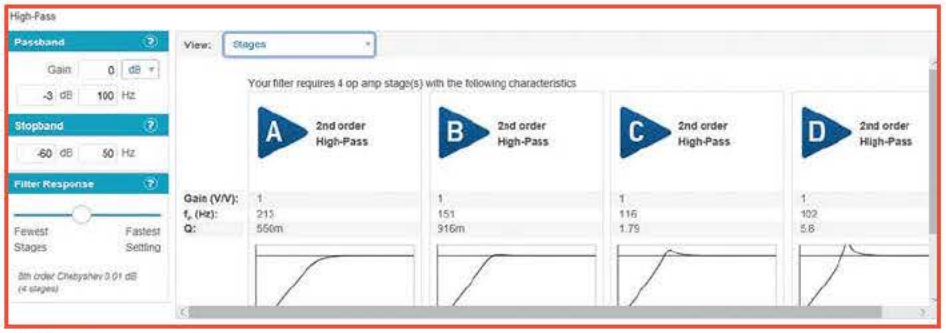

FIGURE 4: Filter wizard high pass filter design info (part).

FIGURE 5: Simulated response of a narrow bandpass filter centred on 550Hz, designed using the online wizard.

The system

The filter module is intended to connect between the receiver's audio output and an active speaker system such as a PC speaker. The device described here operates at audio 'line' levels (around 1 volt); audio from a radio's speaker or headphones output may need the volume turning down in order to prevent overload of the input ADCs on the filter board, which would cause significant distortion.

The signal processor chip chosen for this project is an Analog Devices ADAU1701. There are a few suppliers of boards containing this chip; I bought a package from Sure Electronics [1] that includes the chip on a PCB, along with an audio interface board, a programming board and interconnecting cables for around £50. These three boards are seen in **Photo** 1. The audio interface board can be connected using RCA connectors or stereo jack plugs and operates at line levels. You will need two USB cables for power and data connection and suitable audio connection cables. Analog Devices provides a free software suite called Sigmastudio [2] that massively helps the design process. (Analog Devices also provides a free online filter wizard [3] for designing analogue filter parameters; designs derived from this wizard can provide a useful starting point for some of the figures required when designing a DSP-based counterpart).

Greg Hopkins, G4CST
greghopkins@hotmail.co.uk

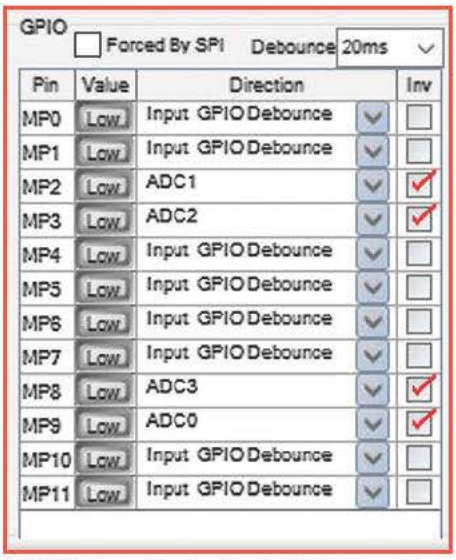

FIGURE 6: Configuring the ADAU1701 registers.

This project uses two of the potentiometers on the processor board to select which of the sixteen low- and high-pass filters are to be used. (An ADC reads the potentiometer position; together, they work as a 16-way switch, though it could just as easily be 4-way or even 64-way). The other potentiometers on the processor board adjust the gain of each channel.

The ADAU1701 has two analogue inputs and four analogue outputs. The audio interface board provides three of the outputs but only one is really needed. Two channels have been used in this project, both of which are passed through the programmable filters.

Programming

Analog Devices' Sigmastudio is a DSP programming software package that takes the complexities of DSP design away and makes design of filters and many other DSP features very accessible to anyone who has a grasp of the basics of signal theory. When you open the software you can drag and drop devices to represent (and simulate) the hardware configuration you're using. In the case of this project, the hardware selected is the Processor chip ADAU1701, the USBi programming interface and the EEPROM memory. Once the hardware configuration is complete, the Inputs, Outputs, Filters and Controls can be dragged and dropped onto the workspace and connections made in order to build the filters.

When you're satisfied that the design is working correctly, the design can be downloaded to the Processor board EEPROM with a single button click. The design is retained in the Processor EEPROM, meaning the system will then work entirely independently of the programming system.

Figure 1 shows the diagram of the project from the schematic window of Sigmastudio. The main elements are the high pass and low pass filter blocks, which are both Index Selectable Multiple band filters. These are grown to sixteen independent filters (R0-R15) and are each four stages long. Each filter from R0 to R15 needs to be individually selected and programmed to achieve the desired response. An example of filter settings is shown in **Figure 2**. The filter setting window is opened by selecting the filter R0 – R15 and then selecting the box in the top right hand corner of the filter block (shift-left click).

Filter parameters are derived from the Analog Devices filter wizard. **Figure 3** and **Figure 4** show screenshots from the wizard design parameters [3] for a 100Hz 4-stage high-pass Chebyshev filter.

From the wizard opening window, select the type of filter you need, set the frequency response and cutoff, then the software automatically calculates the parameters. You could build the filters in hardware (and the wizard even calculates the component values for you). However we don't need to go to that trouble, merely take the values of frequency and Q from the 'Stages' screen and enter them in the filter parameter table in Sigmastudio.

The Stimulus and Probe elements have been added to the design in order to allow the software to show the frequency response of the design. **Figure 5** shows a typical band pass curve when the device is set for CW with a centre frequency of 550Hz.

The AD1701 has four analogue control inputs, Aux_ADC_1 – 4. These are used in conjunction with DC sources and multipliers within Sigmastudio to configure the on-board potentiometers to select individual filters and also to set the volume of each output channel. These must be configured in the ADAU1701 register configuration window as analogue inputs; this is shown in **Figure 6**.

When all the filters have been programmed and the processor configuration settings have been made then the actual device may be programmed. If it doesn't behave as you expected then you can use the powerful simulation features built into Sigmastudio to help debug your design.

The online and help documentation describes all of these features, which are really outside the scope of this article. However, you can keep downloading tweaked configurations to the processor's EEPROM as many times as you like, making incremental changes so the result is closer to your ideal.

Summary

The filter described here is an example of a basic but useful filter for amateur radio applications. I encourage you to change the filter parameters to see what happens; perhaps you'll find a combination more suited to your specific needs. You can download a copy of the .dspproj file for the filter described in this article from the *RadCom* supplementary material section of the RSGB website **[4]**.

There are a myriad of other functions available within the Sigmastudio software, any of which can be added to expand the design. These include (but aren't limited to) things like noise reduction, signal attenuation and gain, signal mixing, bass boost and signal equalisation. Once you've become familiar with the software *[I gained basic familiarity with it much more quickly than I expected, a testament to how well it's written – Ed]*, many features will become apparent that could further enhance the readability signals in particular applications.

Many YouTube videos are online that demonstrate various elements of this sort of system, for example how to connect up the processor board, using the Sigmastudio software and so on. There is also a huge amount of support available from Analog Devices and other sources – including via Wikis, forums and so on – and these are widely-used, well-established products that have a sizeable user base so there will almost certainly be someone 'out there' who can help if you run into difficulties.

I too have made some videos **[5]** outlining the use of Sigmastudio, connecting the DSP hardware and the related programming matters. You may find it useful to start there and move on at your own pace.

Whatever happens, I do encourage you to start experimenting with your own digital signal processing system: it has never been so easy or inexpensive to harness such a fantastic amount of signal processing power for your own applications, not to mention the satisfaction you'll get from mastering some of the latest technology.

Websearch

[1] http://store3.sure-electronics.com/adau1701-audio-digital-signal-processor-kernel-board
[2] https://www.analog.com/en/design-center/evaluation-hardware-and-software/software/ss_sigst_02.html
[3] https://www.analog.com/designtools/en/filterwizard/
[4] www.rsgb.org/radcom
[5] https://www.youtube.com/watch?v=yeM1ijqvTUk&t=478s